# **Итоговый проект**

Для отладки скриптов и запросов я использовал облачный сервис [SQLite](https://sqliteonline.com/) среда выполнения: postgresql

1. **Описание предметной области**

Data\_community – социальная сеть для специалистов в области данных. Пользователи могут доступно обсуждать, читать, размещать посты, реагировать на посты других авторов. Также, каждый участник может получить медали за проявление своих навыков. База данных содержит информацию о пользователях (дата регистрации, очки репутации и т. д.), постах, наградах.

1. **Ожидаемые запросы к предметной области**

* Подсчитать суммарное количество реакций на каждый пост
* Найти самый популярный пост (с наибольшим количеством очков)
* Получение количества постов каждого типа (текстовые, фотографии, видео)
* Получение списка пользователей с их суммарным количеством просмотров профиля

**Изображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание**

1. **Описание базы**

**Таблица** users - cодержит информацию о пользователях.

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| id | int | Идентификатор пользователя, первичный ключ таблицы |
| name | var | Имя пользователя |
| secure\_account | Bool | Защищен ли акант у пользователя  (True, False) |
| last\_access\_date | timestamp | Дата последнего входа |
| creation\_date | timestamp | Дата регистрации пользователя |
| reputation | var | Очки репутации, которые получают за хорошие вопросы и полезные ответы |
| views | int | Число просмотров профиля пользователя |

**Таблица** posts - cодержит информацию о постах.

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| id | int | Идентификатор поста, первичный ключ таблицы |
| title | var | Заголовок поста |
| created\_at | timestamp | Дата создания поста |
| user\_id | int | Идентификатор пользователя, который создал пост, внешний ключ к таблице users |
| post\_type\_id | int | Идентификатор типа поста, внешний ключ к таблице post\_types |
| score | int | Количество очков, которое набрал пост |
| views\_count | int | Количество просмотров |

**Таблица** reaction - содержит информацию о реакциях, это жест, который пользователи ставят посту ((👍=2) или (👎=1))

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| id | int | Идентификатор типа жеста, первичный ключ |
| post\_id | int | Идентификатор поста, внешний ключ к таблице posts |
| user\_id | int | Идентификатор пользователя, который поставил посту голос, внешний ключ к таблице users |
| type | int | Тип жеста |

**Таблица** reward - Хранит информацию о наградах, которые присуждаются за разные достижения.

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| id | int | Идентификатор значка, первичный ключ таблицы |
| name | var | Название значка |
| user\_id | int | Идентификатор пользователя, которому присвоили значок, внешний ключ, отсылающий к таблице users |
| created\_at | timestamp | Дата присвоения значка |

**Таблица** post\_types - Содержит информацию о типе постов.

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| id | int | Идентификатор поста, первичный ключ таблицы |
| type | var | Типы постов \* |

Типы постов \* – посты могут быть таких видов:

1. Текстовые посты = ‘text’
2. Фотографии = ‘foto’
3. Видео = ‘Vid’
4. **Скрипт создания базы**

-- Создание таблицы post\_types

CREATE TABLE post\_types (

id INTEGER PRIMARY KEY,

type VARCHAR

);

-- Создание таблицы users

CREATE TABLE users (

id INTEGER PRIMARY KEY,

secure\_account BOOLEAN,

name VARCHAR,

last\_access\_date TIMESTAMP,

creation\_date TIMESTAMP,

reputation VARCHAR,

views INTEGER

);

-- Создание таблицы reward

CREATE TABLE reward (

id INTEGER,

name VARCHAR,

user\_id INTEGER,

created\_at TIMESTAMP,

PRIMARY KEY (id),

FOREIGN KEY (user\_id) REFERENCES users (id)

);

-- Создание таблицы reaction

CREATE TABLE reaction (

id INTEGER PRIMARY KEY,

post\_id INTEGER,

user\_id INTEGER,

type INTEGER,

FOREIGN KEY (user\_id) REFERENCES users (id)

);

-- Создание таблицы posts

CREATE TABLE posts (

id INTEGER PRIMARY KEY,

title VARCHAR,

created\_at TIMESTAMP,

user\_id INTEGER,

post\_type\_id INTEGER,

score INTEGER,

views\_count INTEGER,

FOREIGN KEY (user\_id) REFERENCES users (id),

FOREIGN KEY (post\_type\_id) REFERENCES post\_types (id));

1. **Скрипты заполнения базы данных**

-- Вставка данных в таблицу “ users ”

INSERT INTO users (id, name, secure\_account, last\_access\_date, creation\_date, reputation, views)

VALUES

(1, 'John Doe', true, '2023-06-01 10:15:00', '2023-05-30 08:00:00', 'Silver', 500),

(2, 'Jane Smith', false, '2023-06-02 14:30:00', '2023-05-31 12:30:00', 'Gold', 1000),

(3, 'Alice Johnson', true, '2023-06-05 18:45:00', '2023-06-01 16:45:00', 'Bronze', 250),

(4, 'Michael Brown', false, '2023-06-03 09:20:00', '2023-06-02 07:15:00', 'Silver', 750),

(5, 'Emily Davis', true, '2023-06-07 12:10:00', '2023-06-03 10:30:00', 'Gold', 1200),

(6, 'Daniel Wilson', false, '2023-06-04 16:55:00', '2023-06-04 14:45:00', 'Bronze', 350),

(7, 'Olivia Johnson', true, '2023-06-05 20:30:00', '2023-06-05 18:20:00', 'Silver', 900),

(8, 'Alexander Smith', false, '2023-06-07 09:05:00', '2023-06-06 07:45:00', 'Gold', 1800),

(9, 'Sophia Davis', true, '2023-06-06 15:45:00', '2023-06-06 13:30:00', 'Bronze', 420),

(10, 'William Johnson', false, '2023-06-07 17:25:00', '2023-06-07 15:20:00', 'Silver', 1100),

(11, 'Ava Wilson', true, '2023-06-01 11:30:00', '2023-06-01 09:15:00', 'Gold', 1400),

(12, 'James Brown', false, '2023-06-02 13:50:00', '2023-06-02 11:40:00', 'Bronze', 550),

(13, 'Isabella Miller', true, '2023-06-03 17:00:00', '2023-06-03 15:00:00', 'Silver', 950),

(14, 'Benjamin Wilson', false, '2023-06-04 08:45:00', '2023-06-04 06:30:00', 'Gold', 1600),

(15, 'Mia Anderson', true, '2023-06-05 10:20:00', '2023-06-05 08:10:00', 'Bronze', 380)

-- Вставка данных в таблицу “ post\_types ”

INSERT INTO post\_types (id, type)

VALUES

(1, 'text'),

(2, 'foto'),

(3, 'Vid'),

(4, 'text'),

(5, 'foto'),

(6, 'text'),

(7, 'Vid'),

(8, 'text'),

(9, 'foto'),

(10, 'text'),

(11, 'Vid'),

(12, 'text'),

(13, 'foto'),

(14, 'text'),

(15, 'Vid'),

(16, 'text'),

(17, 'foto'),

(18, 'text'),

(19, 'Vid'),

(20, 'text'),

(21, 'foto'),

(22, 'text'),

(23, 'Vid'),

(24, 'text'),

(25, 'foto');

-- Вставка данных в таблицу "reaction"

INSERT INTO reaction (id, post\_id, user\_id, type)

VALUES

(1, 1, 1, 2),

(2, 2, 1, 1),

(3, 1, 2, 2),

(4, 3, 3, 1),

(5, 2, 4, 2),

(6, 4, 2, 2),

(7, 3, 1, 1),

(8, 5, 3, 2),

(9, 4, 4, 1),

(10, 1, 3, 2),

(11, 2, 2, 1),

(12, 5, 1, 2),

(13, 4, 3, 1),

(14, 3, 2, 2),

(15, 6, 4, 2),

(16, 5, 1, 1),

(17, 2, 3, 2),

(18, 1, 4, 1),

(19, 6, 3, 2),

(20, 3, 4, 1);

-- Вставка данных в таблицу "reward"

INSERT INTO reward (id, name, user\_id, created\_at)

VALUES

(1, 'Bronze Medal', 1, '2023-06-01 09:00:00'),

(2, 'Silver Medal', 2, '2023-06-02 14:30:00'),

(3, 'Gold Medal', 1, '2023-06-03 11:45:00'),

(4, 'Platinum Medal', 3, '2023-06-04 16:00:00'),

(5, 'Diamond Medal', 2, '2023-06-05 12:30:00'),

(6, 'Bronze Medal', 4, '2023-06-06 08:15:00'),

(7, 'Silver Medal', 3, '2023-06-07 15:30:00'),

(8, 'Gold Medal', 1, '2023-06-08 10:45:00'),

(9, 'Platinum Medal', 4, '2023-06-09 13:00:00'),

(10, 'Diamond Medal', 2, '2023-06-10 16:45:00'),

(11, 'Bronze Medal', 3, '2023-06-11 11:15:00'),

(12, 'Silver Medal', 4, '2023-06-12 09:30:00'),

(13, 'Gold Medal', 1, '2023-06-13 14:00:00'),

(14, 'Platinum Medal', 2, '2023-06-14 12:00:00'),

(15, 'Diamond Medal', 3, '2023-06-15 10:20:00'),

(16, 'Bronze Medal', 4, '2023-06-16 13:30:00'),

(17, 'Silver Medal', 1, '2023-06-17 16:15:00'),

(18, 'Gold Medal', 2, '2023-06-18 11:30:00'),

(19, 'Platinum Medal', 3, '2023-06-19 14:45:00'),

(20, 'Diamond Medal', 4, '2023-06-20 09:45:00');

-- Вставка данных в таблицу “posts”

INSERT INTO posts (id, title, created\_at, user\_id, post\_type\_id, score, views\_count)

VALUES

(1, 'Introduction to SQL', '2023-06-01 09:00:00', 1, 1, 10, 500),

(2, 'Advanced SQL Techniques', '2023-06-02 14:30:00', 2, 1, 15, 1000),

(3, 'Getting Started with Python', '2023-06-03 11:45:00', 1, 2, 20, 750),

(4, 'Data Visualization Tips', '2023-06-04 16:00:00', 3, 2, 5, 250),

(5, 'Machine Learning Basics', '2023-06-05 12:30:00', 2, 3, 8, 800),

(6, 'Introduction to Web Development', '2023-06-06 08:15:00', 4, 3, 12, 1200),

(7, 'Database Optimization Techniques', '2023-06-07 15:30:00', 3, 1, 18, 900),

(8, 'Python Web Frameworks Comparison', '2023-06-08 10:45:00', 1, 3, 25, 1500),

(9, 'Data Analysis with Pandas', '2023-06-09 13:00:00', 4, 2, 7, 350),

(10, 'Introduction to Machine Learning Algorithms', '2023-06-10 16:45:00', 2, 3, 14, 1100),

(11, 'Creating Responsive Web Designs', '2023-06-11 11:15:00', 3, 3, 9, 600),

(12, 'SQL Best Practices', '2023-06-12 09:30:00', 4, 1, 22, 1300),

(13, 'Python Data Visualization Libraries', '2023-06-13 14:00:00', 1, 2, 6, 400),

(14, 'Introduction to Deep Learning', '2023-06-14 12:00:00', 2, 3, 16, 1000),

(15, 'Web Security Basics', '2023-06-15 10:20:00', 3, 3, 11, 700),

(16, 'Optimizing SQL Queries', '2023-06-16 13:30:00', 4, 1, 19, 1200),

(17, 'Python for Data Analysis', '2023-06-17 16:15:00', 1, 2, 4, 250)

1. **Запросы к базе**

* Посчитать суммарное количество реакций на каждый пост

SELECT post\_id, COUNT(\*) as count

FROM reaction

GROUP BY post\_id;

* Найти самый популярный пост (с наибольшим количеством очков)

SELECT title, score

FROM posts

ORDER BY score DESC

LIMIT 1;

* Получение количества постов каждого типа (текстовые, фотографии, видео)

SELECT post\_types.type, COUNT(\*) AS post\_count

FROM posts

JOIN post\_types ON post\_type\_id = posts.id

GROUP BY type;

* Получение списка пользователей с их суммарным количеством просмотров профиля

SELECT users.name, SUM(views) AS total\_views

FROM users

GROUP BY users.name

ORDER BY total\_views DESC;